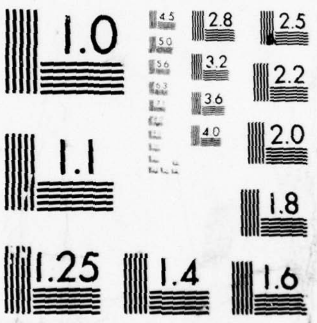
AD-A070 629 DYNAMIC SCIENCES INTERNATIONAL INC SEPULVEDA CA F/G 14/2
TEST PROGRAM SET COST ALGORITHM.(U)
MAY 79 D J ZINGG, A V ROBERTSON, D MCINTYRE DAAB07-77-C-2727

UNCLASSIFIED CORADCOM-77-2727-F-1 NL

1 OF 1

AD
A070629

END
DATE
FILMED
8--79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

LEVEL II

12
B.S.

RESEARCH AND DEVELOPMENT TECHNICAL REPORT
CORADCOM- 77-2727-F-1

TEST PROGRAM SET COST ALGORITHM

D. James Zingg
Al V. Robertson
Dave McIntyre

DYNAMIC SCIENCES INTERNATIONAL, INC.
16611 Roscoe Place
Sepulveda, CA 91343

May 1979

Final Report for Period Sep. 1977 - Apr. 1978

PREPARED FOR:
CENTACS

CORADCOM
US ARMY COMMUNICATION RESEARCH & DEVELOPMENT COMMAND
FORT MONMOUTH, NEW JERSEY 07703

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 19 18 CORADCOM-77-2727-F-1 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 Test Program Set Cost Algorithm.		5. TYPE OF REPORT & PERIOD COVERED Final 7 rept. 21 Sep 1977-Apr 1978 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 D. James Zingg, Al V. Robertson Dave McIntyre		8. CONTRACT OR GRANT NUMBER(s) 15 DAAB07-77-C-2727 NM
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dynamic Sciences International, Inc. ✓ 16611 Roscoe Place Sepulveda, California 91343		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 17 62779A 16 1L762779 AH62 01 011 E23H318 C8 CSCM
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Communications Research and Development Command, ATTN: DRDCO-TCS-MI Fort Monmouth, NJ 07703		12. REPORT DATE 11 May 1979 13. NUMBER OF PAGES 64
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 69p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Test Program Set Cost Elements	Learning Curve
Normalized Cost Curve	Cost Algorithm
ATE Maturity Factor	Life-Cycle Costs
Test System Complexity	UUT Testability
	(Unit under test)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Test Program Set Cost Algorithm provides a methodology for identifying and quantifying the funding costs of major tasks in Test Program Sets (TPS's) development. Areas that are addressed by the study include the learning curve effect, impact of ATE maturity, impact of UUT Testability, effect of design guides on development and life-cycle costs, management-controlled cost factors, utilization of Automatic Test Program (over)

1B

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

411 237

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Generation (ATPG), development costs vs. life-cycle costs, fault insertion and customer "sell-off", and commercial vs. military support. The basic algorithm deals with the analysis, coding, checkout and sell-off of a test program set but other factors such as overhead support, interface device design, ATE compatability, etc., are discussed. The application and useage of the algorithm should provide a valuable estimating tool to assess costs associated with Test Program Set development.

CONTENTS

INTRODUCTION 1

OBJECTIVES 2

THE ANALYSIS 3

Data Base 4
Test Program Set Cost Elements 5
Test Program Development Task Elements 6
Normalized Cost Curve 8
Test System Complexity 16
ATE Maturity Factor 20
Learning Curve. 22

TEST PROGRAM SET COST ALGORITHM 24

THE ANALYSIS (Continued)

UUT Testability 28
Effect of Design Guides on Development and Life Cycle Costs 36
Management Controlled Cost Factors 40
Automatic Test Program Generation (ATPG) 45
Development vs Life Cycle Costs 48
Fault Insertion and Customer "Sell-Off" 50
Commercial vs Military Support 52

TEST CASES 53

SUMMARY/CONCLUSION 55

APPENDIX A 56

APPENDIX B 59

APPENDIX C 63

FIGURES:

1. Twelve-stage Counter with Direct Clear and Output Test Points Selected for Testability 30
2. Examples of Fault Isolation Element Replaceable Component Groups 34
3. Examples of Fault Isolation Element Replaceable Component Groups 35
4. Percent of Functional Test Cost versus Number of Components in Ambiguity Group and Ambiguity Level 39

TABLES:

I. Normalized Costs 14
II. Normalized Cost, Evaluation Sample 15

INTRODUCTION

This document, prepared for the United States Army Electronics Command, Fort Monmouth, New Jersey, under Contract DAAB07-77-C-2727, is intended to serve as an aid in determining definitive cost parameters associated with the development of Test Program Sets (TPS's).

Because of the many tester and UUT testability related variables, costs associated with the Interface Adapter have not been evaluated as part of this study.

Dynamic Sciences International, Inc. (DSII) believes that in light of both DOD and Congressional Actions relative to fiscal year funding, major programs must now fully justify their support costs. Yet, even with supportive rationale, the cost of new program acquisitions has risen to prohibitive sums, with resultant decreases in prime hardware or reduction in support/spares. Additionally, experience during the past two decades has shown dramatic increases in costs of support software, far outstripping the programmed costs for initial prime equipment with operational software.

With the advent of Automatic Test Equipment (ATE), Test Programs and associated interface devices for the testing of complex analog and digital avionics/electronic systems, the government has sought new and innovative ways to establish justifiable costs associated with Test Program Sets.

DSII has compiled a large statistical base of TPS development, and generated guidelines that should provide the United States Army with an estimating tool that will prove efficient and cost effective.

The final intent of this study is to fully define the costs and variables associated with TPS development. This will ensure a final product which meets the fiscal constraints or requests for fiscal budgets where the progress can be effectively monitored against schedule and budget.

OBJECTIVE

The objective of this study is to render to the United States Army an estimating means for funding associated with the development of Test Program Sets under varying conditions and parameters, i.e., complexity of LRU's and SRU's as well as the ATE factors.

The study addresses the following areas:

- o Learning Curve Effect
- o Impact of ATE Maturity
- o Impact of UUT Testability
- o Affect of Design Guides on Development and Life Cycle Costs
- o Management Controlled Cost Factors
- o Utilization of Automatic Test Program Generation (ATPG)
- o Development Costs vs Life Cycle Costs
- o Fault Insertion and Customer "Sell-Off"
- o Commercial vs Military Support

THE ANALYSIS

Dynamic Sciences International, Inc. has reviewed, from its large data base, the salient points listed in the objectives and identified the most critical problem areas for analysis and inclusion into the basic cost algorithm. As each of these areas is examined and discussed, the resultant modifications to the algorithm, if required, or to the total cost parameters will be explained.

The analysis has shown that the most difficult problem in the task of estimating the development of test program sets for electronics is in the identification of complexity and in the anticipation of the factors that will cause development costs to escalate. Decisions made during the development phase can have significant effects on life cycle costs in the maintenance of the electronics. This study has concentrated on the identification of TPS cost elements and establishes an algorithm/equation that can be used to estimate development costs, and discusses the decisions that affect life cycle costs. Although the basic algorithm deals only with the analysis, coding, checkout and sell-off of a test program set, other factors such as overhead support, interface device design, ATE compatability, etc., will be discussed as well. In practical application, these other factors may or may not be applicable to the total cost equation if they are not factors for the specific application. For example, the ATE may be a fully mature system with an operating efficiency of 95% and, therefore, should not be utilized to adjust the total normalized man-hours resulting in a net change to the cost equation.

There is one element that is always difficult to evaluate, and that is the human element; people do not perform technical tasks at the same rate.

The cost algorithm presented here identifies the man-hours required by an average test engineer to design a diagnostic test program set.

Additionally, in order to avoid any problem with ambiguity of terms, each element will be defined as it applies to this study.

DATA BASE

Cost data for the electronic assemblies and their modules (if applicable) was analyzed to generate the conclusions of this study. The electronics assemblies evaluated in this study are listed in Appendix A. Modules evaluated were sampled from S-3A, AAH, Centaur, and B1 electronics. The eight (8) different test systems used in the support of the electronic equipments evaluated were as follows:

VAST AN/USM - 247

HATS AN/USM - 403

Hughes Minuteman II System Test Station

Hughes Minuteman II Computer Test Station

Hughes Minuteman II Platform Test Station

Hughes Minuteman II Gyro Test Station

Teledyne SIU for F-14 CSDC

Teledyne MLT Hybrid Test Set

The cost data base was analyzed to find correlations wherever possible. Relevant experience concerning problems encountered during the development of the data base support programs was used to modify and justify variances when applicable, resulting in many of the other conclusions of this study.

TEST PROGRAM SET COST ELEMENTS

There are a number of elements of cost involved in the development of diagnostic test programs that are considered necessary and supportive in nature. Although the primary goal of this study is to produce a tool for estimating costs associated with the technical aspects of development, a discussion of the other related costs is necessary to arrive at variables that may adjust the overall program cost.

In many instances, other than the design, fabrication and maintenance of the interface devices, most of the support elements required are overlooked. These elements, shown below, must be considered in order that efficient test programming development be accomplished.

- Maintenance and calibration of test equipment.
- Inventory control of UUT assets.
- Technician support for UUT maintenance, and preventative maintenance of Interface Adapters.
- Keypunch/verification services.
- Operation and maintenance of compilation facility.
- UUT configuration tracking/control.
- Clerical support.

The cost of these support items, while not identified by percentage, can vary from contractor to contractor.

TEST PROGRAM DEVELOPMENT TASK ELEMENTS

The breakdown of the test program development task elements was accomplished using the cost data base for three (3) large-scale development programs. Since the data base was large, many different types of electronic assemblies and modules were involved, representing many different complexities. The development task percentages represent the average of all complexities. When applying the algorithm to a single item of electronics, these task element percentages will vary as a function of the complexity and function of the specific electronics. These differences will be discussed in the presentation of the normalized cost curve.

The study results depict an average percentage breakdown of the development tasks as follows:

Circuit Analysis -	25%
The analysis of the unit to be tested for functional and diagnostic test formulation.	
Test Design -	12.5%
The application of the test formulation to a specific item of test equipment.	
Code and Compile -	7.5%
The generation of an object program to be executed on the test equipment.	

Functional Test Integration - 15%

The time required to prove that the test program will accept a good electronic component and reject a failing one. This percentage includes both on-station debugging time and off-station program rework time.

Diagnostic Test Integration - 30%

The time required to verify the accuracy of the test program diagnostics. On-station debugging and off-station program rework time are included.

Validation/Sell-Off - 10%

The formal demonstration of the test program set to the customer includes the on-station demonstration time and a review of all deliverable test program documentation.

These percentages are valid for the normalized cost curve and will vary only as other factors are considered in the algorithm. Basic assumptions made in order to arrive at these percentages were as follows:

o Circuit Analysis and Test Design -

The following data was available:

Detailed schematic or logic diagrams (with reference designators); top assembly drawing (with connector part numbers and dimensional definition); component specifications; and module interconnect information for electronic assemblies.

The UUT was available for visual inspection.

- o Code and Compile -

  High level language such as ATLAS with test station on-line edit capability. Off-line compilation must also be available.

- o Integration -

  At least two working UUT's are available at all times.

NORMALIZED COST CURVE

The normalized cost curve is the key element of the algorithm and is the basis for estimating test program set development. It assumes total efficiency and an optimum environment for the development of test programs by a test engineer.

Assumptions made in establishing the normalized costs were as follows:

- o UUT was ATE testable.
- o Test engineer had working knowledge of diagnostic strategy.
- o Test engineer was experienced on the test equipment.
- o Test engineer performed the complete test programming function, i.e., circuit analysis and establishment of the test strategy and all other tasks through formal sell-off.
- o All peripheral factors such as test equipment availability were reduced to the zero obstacle level - engineering man-hours under ideal conditions.
- o Probing access during test program integration. (Accessibility)

- The test equipment met the test requirements of the electronics to be tested.

- Test equipment had on-line edit and compilation capability, but separate off-line compilation facilities were also available.

Experience has shown that the most accurate method of estimating any computer software effort is to estimate the lines of code that must be generated to perform the task. In the field of diagnostic software for electronics, this can ideally relate to the number of tests* that will be performed.

However, since the number of tests required cannot be identified until the test program has been designed, a suitable alternate had to be found that could be utilized as a basis for comparison.

This alternative, forming the basis of the algorithm,is the number of pins available to the test equipment, modified by a complexity factor, resulting in a total number of man-hours required to do the task. For the time related factors (ATE maturity and learning curve), the man-hours must be converted to a meaningful calendar schedule. Upon completion, the time factor delta will be added to the original man-hour estimate.

In addition to estimating the man-hours required, the algorithm will also produce on-line test station hours required to do the program development task.

For example, a pure digital electronics module that can be tested statically** represents the lowest level of complexity. Stimulus and responses are

*A test is defined as the application of a stimulus and the evaluation of a response using the same test equipment hookup.

**Static testing as defined here is digital testing at other than operational frequencies, monitoring only the logic states at each clock.

monitored on a one-for-one basis using the same test equipment hookup. A count of the number of pins is a measure of the number of tests to be performed. Certain parameters such as number of power pins or circuit redundancies will be discussed later.

Another example of complexity as used in the algorithm lies in the area of RF testing. In most instances, many types of measurements will be made using the same stimulus setup with different response measurements. One is interested in not just the presence or absence of a signal but the quality of many functional signal parameters.

Thus, for purposes of using this algorithm, complexity is defined and treated as families of types of electronics, relying on their similarities as a group, to estimate the cost of doing a test program set. The electronic families identified in this study that have unique test characteristics or requirements are as follows. Some of the categories have been identified only for their station hour requirements which has been a by-product of this study.

- o RF Electronics - This family is defined as an electronic assembly that can detect or generate a modulated signal.

- o Microprocessor Electronics - An electronic assembly that contains a "hard-memory" microprocessor.

- o Real-Time Digital Electronics - This family includes any digital device that must be tested at its functional operational clock frequency.

- o Converter/Interface Electronics - An electronics assembly which reformats parameters in specialized form for output distribution to other devices.

- o Manual Interface Electronics - Electronics assemblies which require some sort of manual intervention during test. This action can be button pushing, switch setting, potentiometer adjustment, etc. <u>It is important to note that in the counting of interface pins, the manual actions must be counted, i.e., a three-position switch represents three (3) interface pins.</u>

- o Static Non-parametric Testing - Electronics that are not tested at operational frequencies. Test involves checking only for presence or absence of signal common in digital modules, switching units and signal distribution units.

- o Other - The balance of electronics systems evaluated does not seem to present any particular characteristics that would cause costs to vary. Cost data variances did not seem to be significant in a group that included power supplies, electro-mechanical, indicators, monitor and control, and other single function electronics.

As was previously mentioned, these categories were created as a result of a significant difference in either development or station time requirements. Table I lists the basic costs in man-hours and station hours for each of the electronic types cited. Table II lists the population of electronics studied for each classification type.

As mentioned earlier, the different families of electronics because of their design can cause variances in the development task element percentages. These changes will be caused by the lack of accessibility and the impact on integration, and vary in each category. The accessibility factors are shown in Table I, and should be used when estimating LRU's in categories 1, 2 and 3. Modules (SRU's) should be evaluated for their accessibility on an individual basis. LRU's in categories 6 and 7 should be evaluated for their accessibility on an individual basis.

Descriptions of the Normalized Costs by category are as follows:

1. RF - The most costly development category per pin. Loss calculations, selection of test tolerances and the number of measurements to be made are the primary reasons for the high cost. Accessibility is hampered by canned/shielded multi-component circuits, waveguide and coax connectors and in general the inability to gain access to critical circuit nodes during checkout. LRU access is normally a problem and the integration and test station time should be doubled, an accessibility factor of 1.

2. Microprocessors - As defined herein, the functional test will consist of executing a "firmware" program. This type of testing requires the gathering of data in real time without operational clock control (Diagnostics become an analysis of the gathered data.). Circuit card extension is virtually impossible to preserve timing integrity and probing is generally useless because of execution times. This lack of accessibility will cause 100% increase in test station and integration development hour requirements, therefore the accessibility factor is 1.

3. Real Time Digital - Real time integration problems coupled with non-extendability of circuit cards and problems associated with probing can cause test station and integration hours requirements to increase by 67%. (Accessibility factor of 0.67)

4. Converter/Interface - The definition of this function does not present any abnormal accessibility problems.

5. Manual Interface - This category does not contribute to accessibility problems.

6. Static/Non-parametric - Certain UUT's in this category will have access problems and should be analyzed. If access is a problem, use an accessibility factor of 1.

7. Other - Since this category has the largest electronic mix, each UUT should be evaluated for accessibility and probing. Power supplies might pose safety problems and some electro-mechanical devices are often sealed units. If access is a problem in this category, increase test station and integration hours by 50%, an accessibility factor of 0.5.

TABLE I

NORMALIZED COSTS

Category	Electronic Type	Test Station Hours/Pin	*Engineering Development Hours/Pin	Accessibility Factor
1	RF	4	40	1.0
2	Microprocessor	3.5	35	1.0
3	Real Time Digital	3	30	0.67
4	Convertor/Interface	2	20	---
5	Manual Interface	1	10	---
6	Static/Non-parametric	0.5	5	1.0
7	Other	2	20	0.5
8	Redundancies	1	10	---

*For functional test estimates, count only the functional pins.

TABLE II

NORMALIZED COST

EVALUATION SAMPLE

Category	# of LRU's Assemblies	# of SRU's
RF	8	45
Microprocessors	3	0
Real-Time Digital	12	1
Interface/Converters	13	0
Manual Interface	8	1
Static Digital	0	67
Other	22	43

For the purposes of creating the normalized cost, remember that many electronic assemblies will require the use of more than one category. This was true in many of the electronics studied such as manual interface electronics that were initialized, controlled and/or monitored via computer (real-time) control. Analysis time spent in classifying the electronics along with their associated pin counts will ensure a more accurate and reliable cost estimate.

In estimating costs associated with the testing of electronics with redundancies, the redundancies must be separated from the pin count and treated separately using one station hour per pin and 10 man-hours per pin. The rationale provided is that while the basic electronics might be part of a unique family, the actual work is a duplication of other work and falls into the smaller cost range.

An additional factor that will be harder to recognize from actual schematic data is in the identification and treatment of real-time serial digital interfaces. Serial digital interfaces should be considered as having a pin for each bit of data in the serial word.

In order to better understand the intricacies of the algorithm, the estimates for the example electronics supplied by ECOM will be covered in a step-by-step manner in the latter part of this study.

TEST SYSTEM COMPLEXITY

Experience has shown that from one major program to another, the test equipment and its inherent complexity has played an important role in contributing to test program development costs. Since test systems are usually made up of a combination of stimulus and response instruments, whose functions are known and familiar to the test engineer, the total system with its inherent packaging and switching was used in determining a complexity factor. As the size of the interface grows, the switching design becomes more important and of necessity more complex. The complexity factor was generated and is used in both the test equipment maturity and learning curve equations. Although the nature of the complexity factor may appear arbitrary, it is made up of elements that contribute to test system complexity and produce a consistent offset for the empirically derived curves.

The complexity factor (K) derived is as follows:

$$K = \frac{100f}{\sqrt{R \times I}}$$

where I represents the switching interface size in pin count,
f represents the number of independent test functions, and
R is an actual count of the number of racks of equipment in the system.

All racks should be counted, but duplicate or redundant functions should only be counted once. The smaller the K factor, the more complex the system because of its relationship in the efficiency equations. For single function testers, let I = 2.

The K factor ranges for the test systems evaluated in the study were from 13 to 47.

Using the equation, the complexity factor (K) for AN/USM-410 (EQUATE) would be as follows:

functions(f) = 23

1. AC Standard
2. DC Standard
3. AC Power Supply

4, 5, 6, 7, 8 DC Power Supplies

9. AFG
10. Digital Input
11. Digital Output
12. Synchro

Sample Measurement Subsystem

13. Low Speed Voltage Sampler
14. High Speed Voltage Sampler
15. Frequency Sampling Unit
16. RF Stimulus

RF Responses

17. Power Measurement
18. Frequency Measurement
19. Impedance Measurement
20. Network Analysis
21. Spectrum Analysis
22. Switching
23. RF Switching

Racks (R) = 6

Interface(I) = 930

$$K = \frac{100(23)}{\sqrt{6 \times 930}} = 30.9$$

Two examples of functions would be as follows:

1. A programmable D-C power supply is one source function.

2. An arbitrary function generator (common in 3rd generation testers) is also counted as one function.

ATE MATURITY FACTOR

Since the advent of Automatic Test Equipment, developers of Test Program Sets have found that the maturity of the system is a significant cost driver in the total development costs of TPS. The introduction of any new system requires a period of time for shakedown and modification before the system reaches its specified availability. In parallel, and subsequent to delivery, documentation and training must follow in a normal progression in order that the desired degree of efficiency be obtained.

Once a commitment has been made for the procurement of a particular test system, the inherent responsibility is to make that system perform to its specification, and to its specified efficiency, within the shortest possible time frame.

Just as in the case of the learning curve, the ATE maturity factor contributes to the various elements of cost growth and must be included in the total costs of TPS development.

The efficiency equation that has been evolved for test equipment maturity (TEM) uses the test station complexity factor, previously discussed, with the time calculated in calendar months.

The maturity equation is:

$$\text{TEM Efficiency } (t_x) = \frac{100t^2 + (7.08K - 220.8)t + (0.15K^2 + 24.5K)}{t^2 + (0.0708K - 2.208)t + (0.15K + 24.5)}$$

where t = calendar months from t_0, t_0 being the first operational month of the first test system

and K = test equipment complexity factor

$$\text{TEM Man-hour} = \sum_{t}^{t_n} (100 - \text{TEM Efficiency } (t_x))\ 1.66$$

where $t_n = t +$ the predicted station months using the normalized base calculated from Table I

Calculating the TEM efficiency for EQUATE yields the following efficiencies, reaching full maturity in 1976.

TEM Efficiency	Months from first operational system
33.3%	1
39.3%	2
47.2%	3
55.4%	4
62.8%	5
69.1%	6
78.4%	8
84.4%	10
88.4%	12
92.9%	16

LEARNING CURVE

Two learning curves were identified: namely, the individual's experience as a test engineer, and the learning curve (LC) for the ATS itself. Both curves are efficiency curves and will result in increased man-hours for the task as they apply. The individual experience curve could not be derived from the data base and is not included. Experience with test engineers over a period of 15 years has made the inclusion of this discussion relevant to this study.

The individual experience curve applies only to the test engineer. Since there are a multitude of disciplines involved in the generation of a test program, the individual has a learning curve. Actually, each of the tasks involved in the development of a test program set has a learning curve, but for the purpose of this algorithm it was treated as a single task. There have been a number of career casualties in this field, and there are some that perform far better than the data would indicate. The majority of participants in the test program developments examined were experienced test engineers. For purposes of this study, a test engineer with two years of experience in the analysis, code and compile and integration disciplines should be considered at the peak of his efficiency and effectiveness.

The learning curve (LC) that is an important part of this algorithm is the test system learning curve. This curve represents the efficiency of the individual on a given piece of test equipment. The peak system efficiency will be attained at different times for different test systems, depending on their complexity. This factor is added to the base algorithm hours for each program as it applies. The learning curve equation, generated from empirical cost data, is as follows:

$$\text{LC Efficiency } (t_x) = \frac{100t^2 + (19KM + 585)t + (40.5KM + 2835)}{t^2 + (0.161KM + 5.65)t + 81}$$

where t = calendar months experience on the system by the engineer,

$$\text{and } M = 1 + \frac{\text{TEM Efficiency}}{90},$$

and K = test equipment complexity factor.

This learning curve was based on data from two new test systems that were not mature, and maturity is a major factor in the learning curve. As a test system reaches full maturity, the learning of that system becomes easier because personnel that have reached their learning curve peak on that test system are available for employment, consulting, training, etc.

$$\text{LC Man-hour} = \sum_{t}^{t_n} (100 - \text{LC Efficiency } (t_x))\ 1.66$$

where t_n = t + the predicted station months using the normalized base calculated from Table I.

The calculated learning curve for the mature EQUATE test system is as follows:

Calendar Months on Station	LC Efficiency %
1	73.8
2	79.8
3	84.3
4	87.8
5	90.5

TEST PROGRAM SET COST ALGORITHM

Although other factors will be discussed in later portions of the text, the complex factors of the algorithm have been defined and explained. Because of its complexity, the algorithm as delivered is a FORTRAN program. To give a better understanding of the algorithm, it is depicted in flowchart format on the following pages.

The data to the input to the FORTRAN program is as follows:

CARD 1

	Card Column 1
Cost Estimate total test program	1
Cost Estimate TRA only	2
Cost Estimate TPS minus TRA (different supplier)	3
Cost Estimate TPS minus TRA (same supplier, different engineer)	4
	Card Column 3

Does Test System have on-line edit?	yes = 1
	no = 0

	Card Columns 5,6
Complexity Factor (K) of Test System	xx
	Card Columns 8,9
Time in months (t) for TEM	xx

CARD 2

Categories, pin count and accessibility separated by commas from Card Column 1. Example from the test case, page 53, if electronics are accessible use 1; if not use 0.

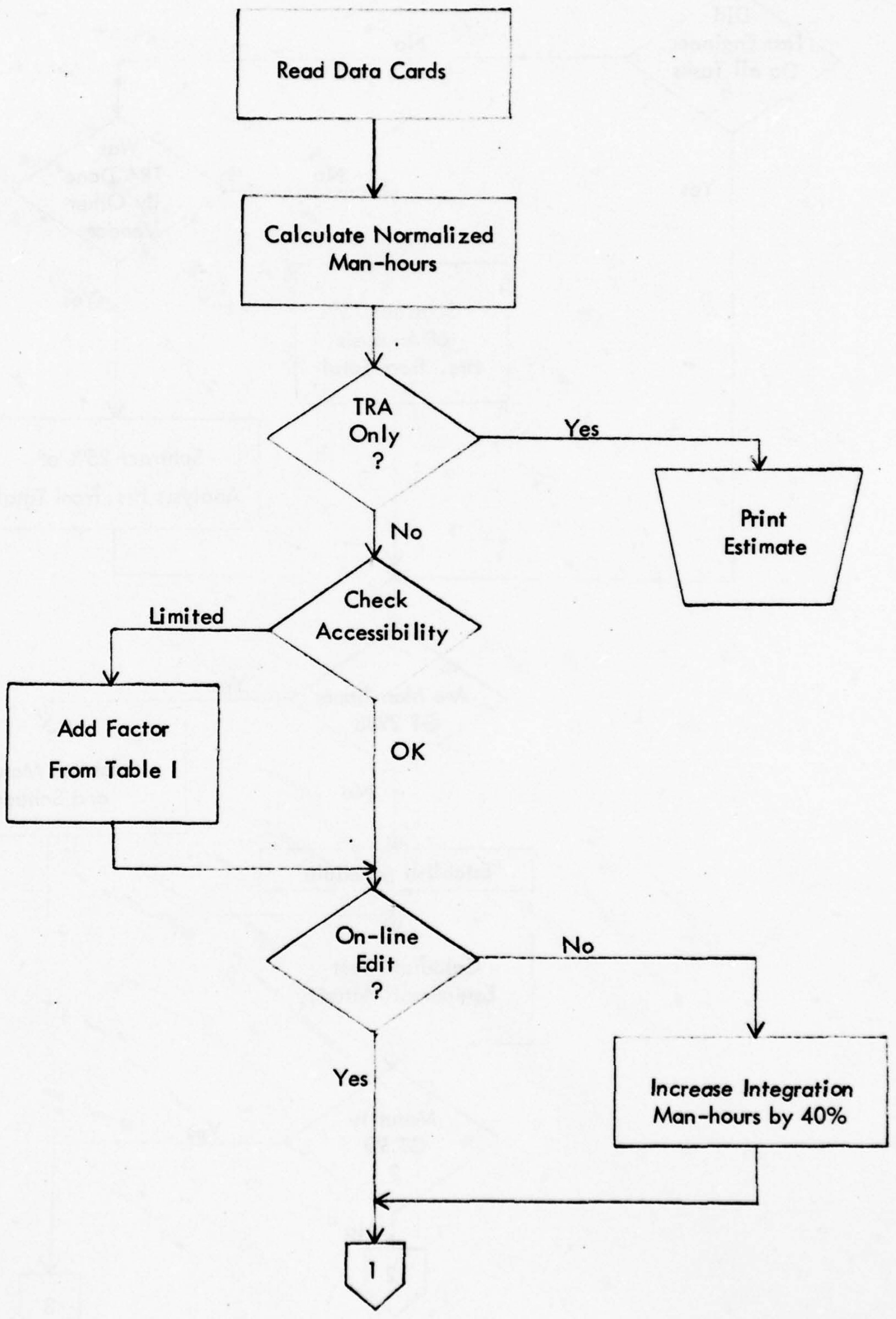
1, 017, 0, 5, 012, 1, 7, 040, 1

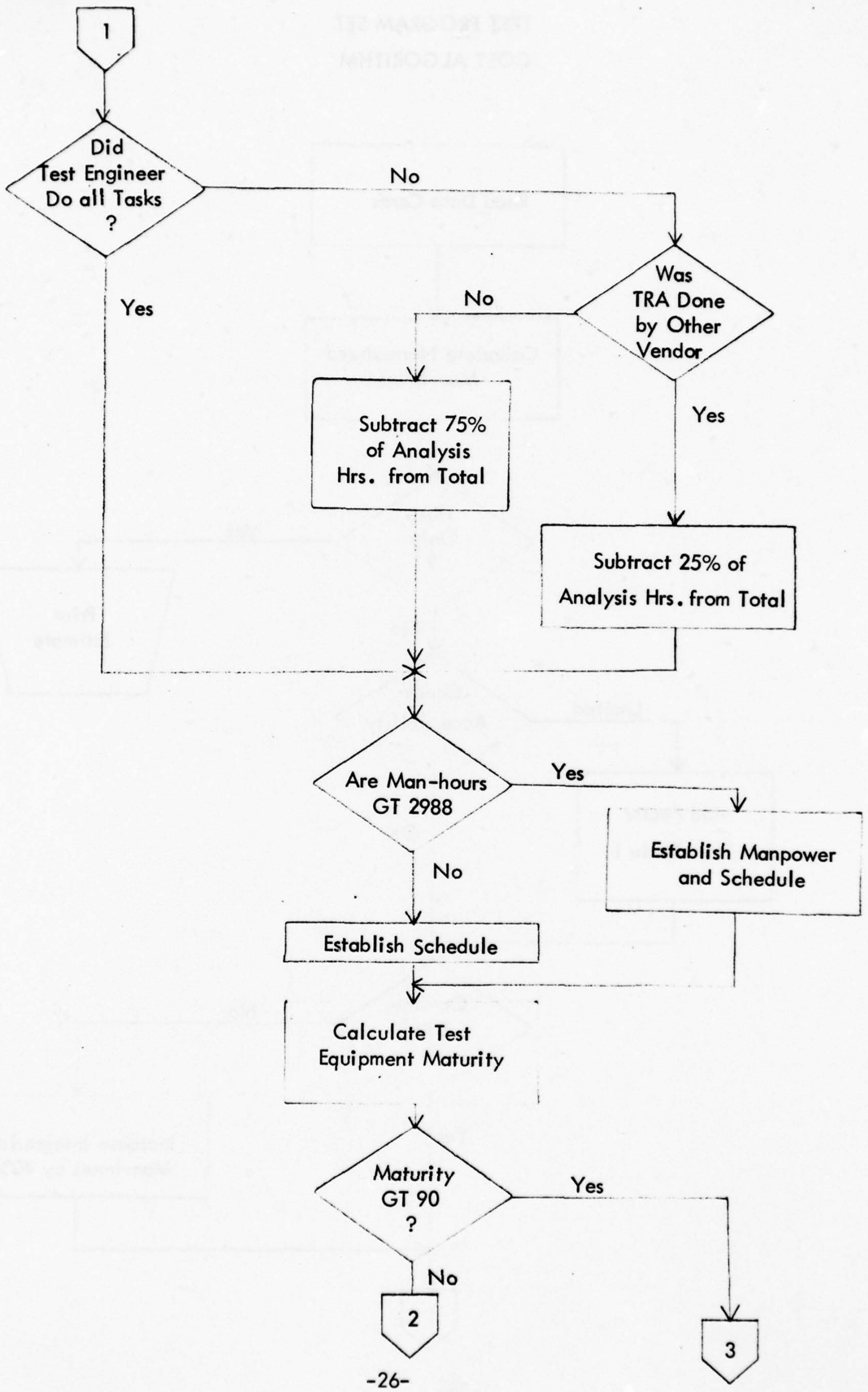
CARD 3

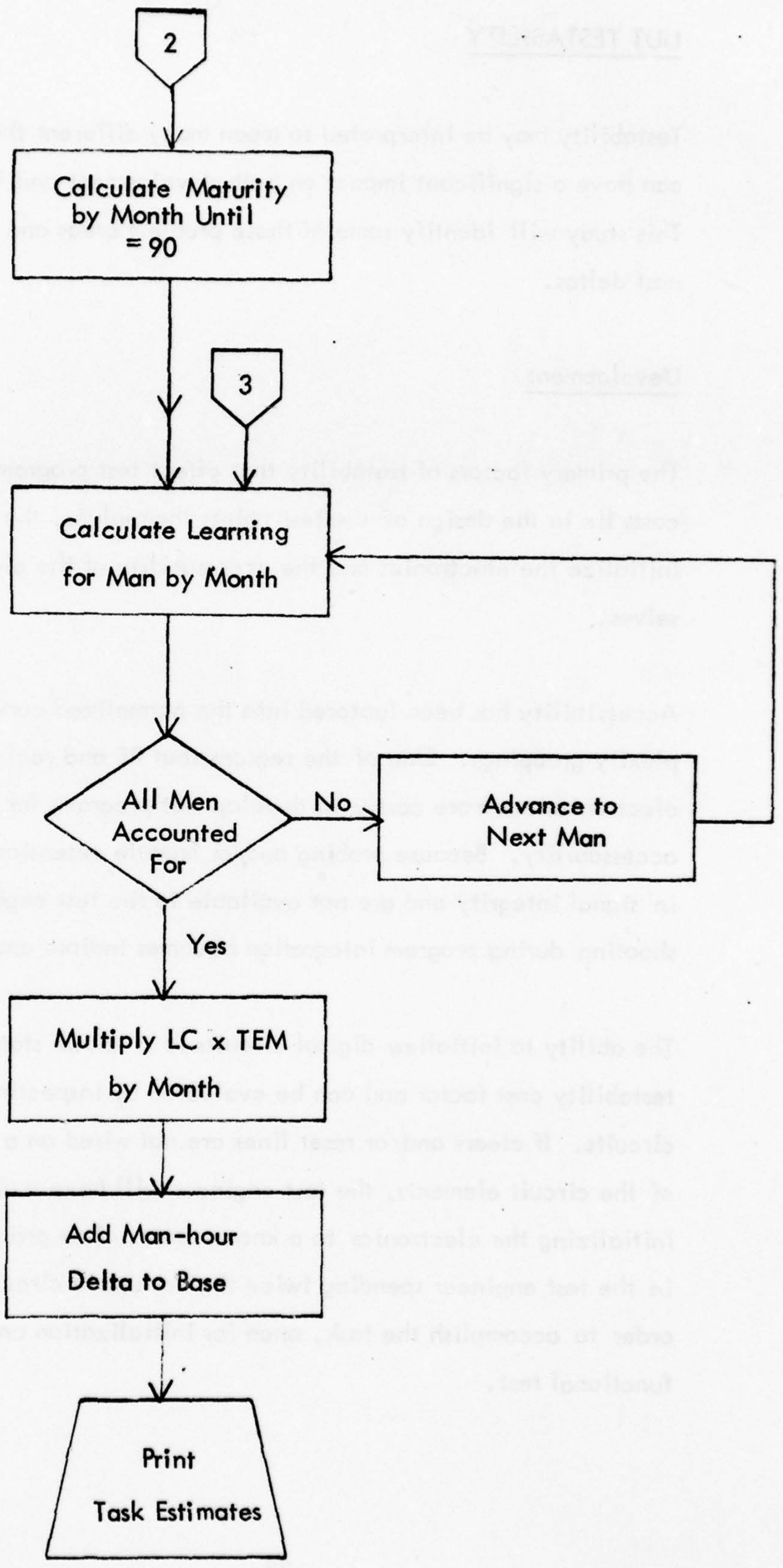
Engineers available to perform the task, with number of months experience on the Test System. Start in Card Column 1.

17, 9, 3, 0, etc.

TEST PROGRAM SET
COST ALGORITHM




1
Did Test Engineer Do all Tasks ?
No
Yes
Was TRA Done by Other Vendor
No
Yes
Subtract 75% of Analysis Hrs. from Total
Subtract 25% of Analysis Hrs. from Total
Are Man-hours GT 2988
Yes
No
Establish Manpower and Schedule
Establish Schedule
Calculate Test Equipment Maturity
Maturity GT 90 ?
Yes
No
2
3


2
Calculate Maturity by Month Until = 90
3
Calculate Learning for Man by Month
All Men Accounted For
No
Advance to Next Man
Yes
Multiply LC x TEM by Month
Add Man-hour Delta to Base
Print Task Estimates

UUT TESTABILITY

Testability may be interpreted to mean many different things - all of which can have a significant impact on both development and life cycle costs. This study will identify some of these problem areas and their predicted cost deltas.

Development

The primary factors of testability that affect test program set development costs lie in the design of the test points themselves, the ability to easily initialize the electronics and the accessability of the electronics themselves.

Accessibility has been factored into the normalized curve with the complexity groupings. One of the reasons that RF and real-time digital electronics are more costly to develop test programs for is the lack of accessability. Because probing and/or module extension cause problems in signal integrity and are not available to the test engineer, troubleshooting during program integration becomes tedious and expensive.

The ability to initialize digital circuits to a known state is an important testability cost factor and can be evaluated by inspecting the actual circuits. If clears and/or reset lines are not wired on a large majority of the circuit elements, the test engineer will have a difficult task in initializing the electronics to a known state. This problem will result in the test engineer spending twice the amount of circuit analysis time in order to accomplish the task, once for initialization and once for the functional test.

Test Points

The proper selection of test points can have a direct effect on the cost of test program development. These costs can be seen in the areas of test analysis and in the interface device design and development.

In the area of test analysis, the proper placement of test points will reduce the amount of time required by the test engineer to determine the initial conditions of a circuit both by monitoring intermediate outputs, and by allowing external control of storage elements that in a system environment are in a known or "don't care" condition, but must be "forced" in the test environment.

The example depicted in Figure 1 shows three (3) 4-bit binary counters connected to form a 12-stage counter with a maximum count of 4096. The test points in this example are placed on the direct clear lines and on the outputs of each of the 4-bit stages. When power is applied to the circuit, the state of the counter is unknown. The test points on the direct clear lines allows the counter to be reset to a known condition without counting through a possible 4095 counts. This results in a savings of program development time and in test time. The test points on the outputs of each of the 4-bit stages are primarily for fault isolation. If a fault is detected at the output pin, it cannot be determined which of the three stages is not functioning properly without the ability to monitor an intermediate point. The test points on the output of each stage allows these points to be monitored, which results in faster and more complete isolation.

The cost of interface adapter design and development is also effected by the test point selection. The addition of test points to noise sensitive lines such as direct set and resets of flip-flops, one-shot inputs and flip-flop outputs makes it necessary to provide buffering in the interface device.

The buffering must be provided at the closest possible point to the unit under test to prevent the pickup of unwanted signals that can cause erratic operation of the device being tested. The buffers can be provided on the UUT, thus eliminating the requirement in the ID; but if they are not, the impact on test program development must be considered.

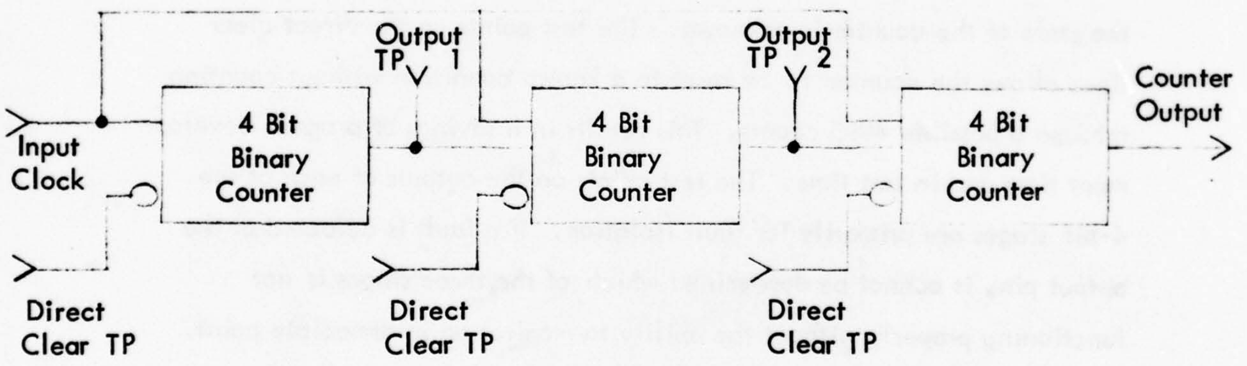


FIGURE 1

TWELVE-STAGE COUNTER WITH DIRECT CLEAR AND OUTPUT TEST POINTS SELECTED FOR TESTABILITY

Ambiguity/Isolation

Diagnostic isolation capability and ambiguity are controlled by the electronics design engineer. If he uses good testability design guidelines, isolation can be accomplished. The test program cannot make up for shortcomings in the design and cannot be effective without meaningful, well-designed test points. This problem can be solved with the use of well-written, enforced testability guides. Test programs for electronics with adequate test points for unambiguous isolation of failures will cost more in the development phase.

A great deal of research was spent in this study in an attempt to identify and define a quick measure of the testability of electronics by using statistical criteria. Because of the variations in packaging and in the arbitrary, almost random selection of test points in the electronics evaluated, a statistical measure of testability could not be established.

Life Cycle

Testability factors that have impact on life cycle costs include module keying, connector commonality and isolation ambiguity.

If keying for modules with identical connectors cannot be handled in the interface Adapter design, or by simple maintenance actions on the part of the test operator, more interface adapters will be required. As a result, this will have a large and significant effect on the costs involved in the activation of any new maintenance site. Keying requirements for military electronics should be re-evaluated. The "fool proof" thought of placing the wrong module in the wrong slot is reasonable for adjacent locations, but entirely cost defeating if required throughout the whole electronics assembly.

The wide variety of connectors encountered in electronic assemblies also contributes to the high life cycle costs in activating maintenance sites. Therefore, a recommendation resulting from this study is that a small family of connectors should be specified for electronics design. This would result in a decreased number of interface adapters that would be required, thereby lowering life cycle costs.

The impact of test program isolation ambiguity on life cycle costs is not as obvious as the previous examples, and yet may be a larger problem. Isolation ambiguities present problems in logistics sparing and repair of the electronics. A successful maintenance action calling for the replacement of two modules will in reality place one good module in the test and repair cycle. If the modules are replaced one at a time as part of the maintenance action, the labor and test station time requirements are increased an average of 25%. In either instance, the overall effect results in a reduction of maintenance effectiveness at an increased life cycle cost.

The problem of isolation ambiguity at the module level of test and repair can create even a greater cost problem. Diagnostic piece part isolation at the module level is an incredibly expensive task and should not be considered as a solution to this problem. Data evaluated in this study indicates that logistics sparing be based on families of components that will, of cost necessity, be the lowest level of isolation. This can be accomplished by establishing element groups at the outset of circuit analysis. Two examples of this approach are shown in Figures 2 and 3. If sparing is accomplished using the component group concept, repair costs would be greatly reduced.

Unambiguous isolation at the assembly level of test presents a different set of problems, and multiple module sparing is a very expensive undertaking. Where diagnostic ambiguities cannot be accomplished automatically, manual procedures should be provided to successfully isolate to one module.

Although the analysis and publication costs will be affected, life cycle cost savings will be achieved in both spares requirements because of the time improvement in the repair cycle.

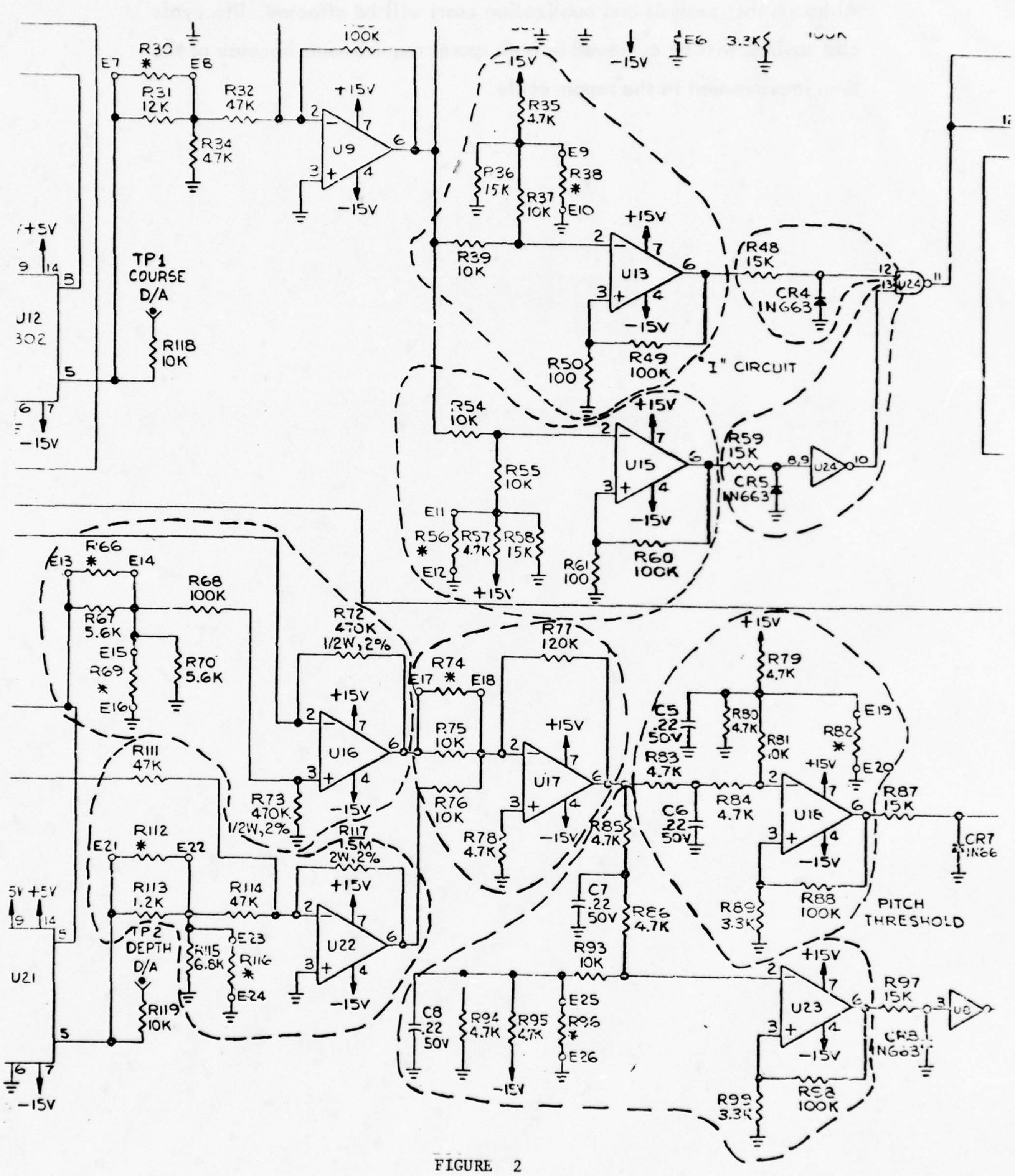


FIGURE 2

EXAMPLES OF FAULT ISOLATION ELEMENT REPLACEABLE

COMPONENT GROUPS

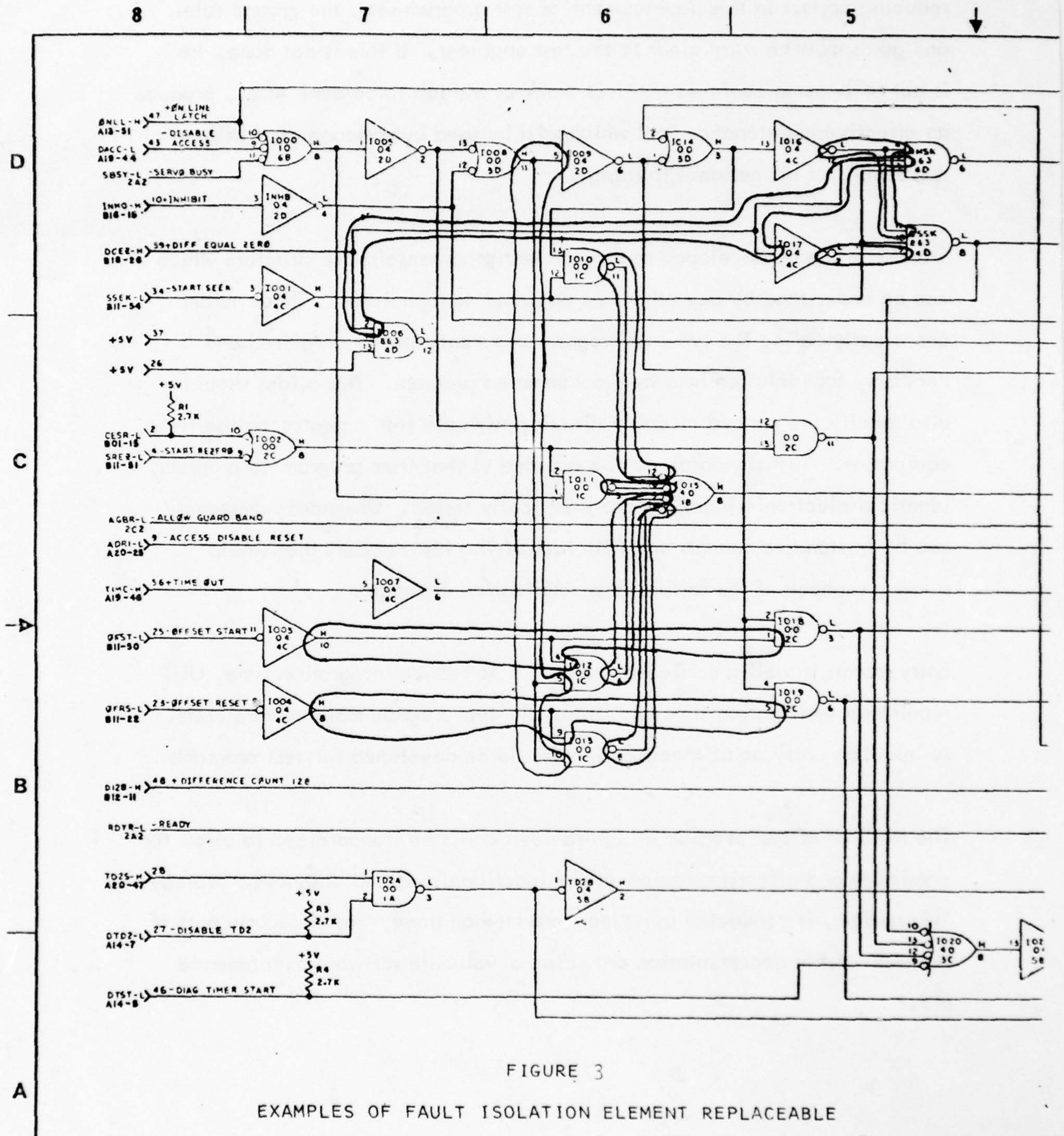


FIGURE 3

EXAMPLES OF FAULT ISOLATION ELEMENT REPLACEABLE

COMPONENT GROUPS

EFFECT OF DESIGN GUIDES ON DEVELOPMENT AND LIFE CYCLE COSTS

The importance of design guidelines is often overlooked as a method of reducing costs. In this development of test program sets, the ground rules and goals must be very clear to the test engineer. If this is not done, he is apt to become confused and lose track of the job to be done (i.e., produce an effective maintenance tool which will be used by someone els e without the benefit of his personal training).

Standards can be developed to force a test program software structure which can be understood by any other test engineer or technician. These guides can clearly define the types of program comments that are required and necessary for inclusion in a well documented program. The guides should also specify the method of communicating with the test operator on the test equipment. Test standards can be adopted so that from program to program, identical electronic functions are identically tested. Diagnostic messages can be in standard formats and with identifying test numbers that would allow a recheck of the failure or a verification of repair.

Entry points, if well specified, can do much to reduce integration time, UUT repair and verification time and life cycle test program maintenance costs. An explicit entry point specification should be developed for test programs.

The methods of test program design reviews could be standardized to allow for consistent and effective review of the functional test and diagnostic strategy. This review, if conducted using standard methodology, could become part of the deliverable documentation providing a valuable software maintenance aid.

Since test languages have all but been standardized into very readable English, a well structured and annotated program would be the only software maintenance documentation required other than the electronic drawings. An established test program structure and test procedures would also do much in reducing software maintenance costs. Many errors have been made, and dollars spent, in the correction of defects/errors in delivered computer software.

Test program sell-off, and establishing the demonstration criteria has long been a time consuming and costly procedure. The measurement of diagnostic isolation ambiguity has never clearly been defined. A quote directly from NAVAIR document AR-10A is as follows:

"In at least 90 percent of the cases of probable malfunction of an SRA, the fault shall be isolated to that sole SRA."

The above statement of test program design criteria by the government and the contractor are usually misconstrued and result in contractual disputes. The primary cause of the dispute is the method of measuring the test program against the ambiguity requirements.

A sample of faults selected for a demonstration cannot be expected to represent an accurate reflection of the isolation ambiguity for that TPS, only 100% insertion of faults will result in an accurate measure. When dealing with small samples of faults, it is relatively easy to select a set of faults that will not meet the contractual isolation requirements. This can be done on LRU's and SRU's where complete fault insertion would result in a totally compliant TPS. Disputes resulting from this type of situation result in higher costs and seldom improve the real quality of the TPS.

A design criteria and a method of measuring the test program's performance to that criteria should be established. In this manner, costs in the analysis and sell-off phase can be predicted and controlled.

Isolation ambiguity costs have been studied for SRU's (modules) and are depicted in Figure 4. As shown, single component isolation can be very costly. The data from which the curves were derived was gathered on modules where probing and lead lifting was allowed and access was not a problem. This data is for modules, and could be used to calculate life cycle component group sparing versus costs of diagnostic isolation to different component levels. The isolation element groups (shown in Figures 2 and 3) represent the recommendations of this study, and the cost intercept is shown in Figure 4.

These curves (Figure 4) are again based on average component isolations and cannot be used as a guide for every module. The analog module (Figure 2) and the digital module (Figure 3) have average isolation element groups of 6.7 and 2.6 respectively, and this can be expected across a population of modules. The isolation levels will always vary, and the method of isolation ambiguity measurement is still a problem.

Isolation ambiguity at the component level should have basic guidelines, with the understanding that each particular module is evaluated on its own from a cost and maintenance viewpoint.

SRU TPS AMBIGUITY/ COST ANALYSIS

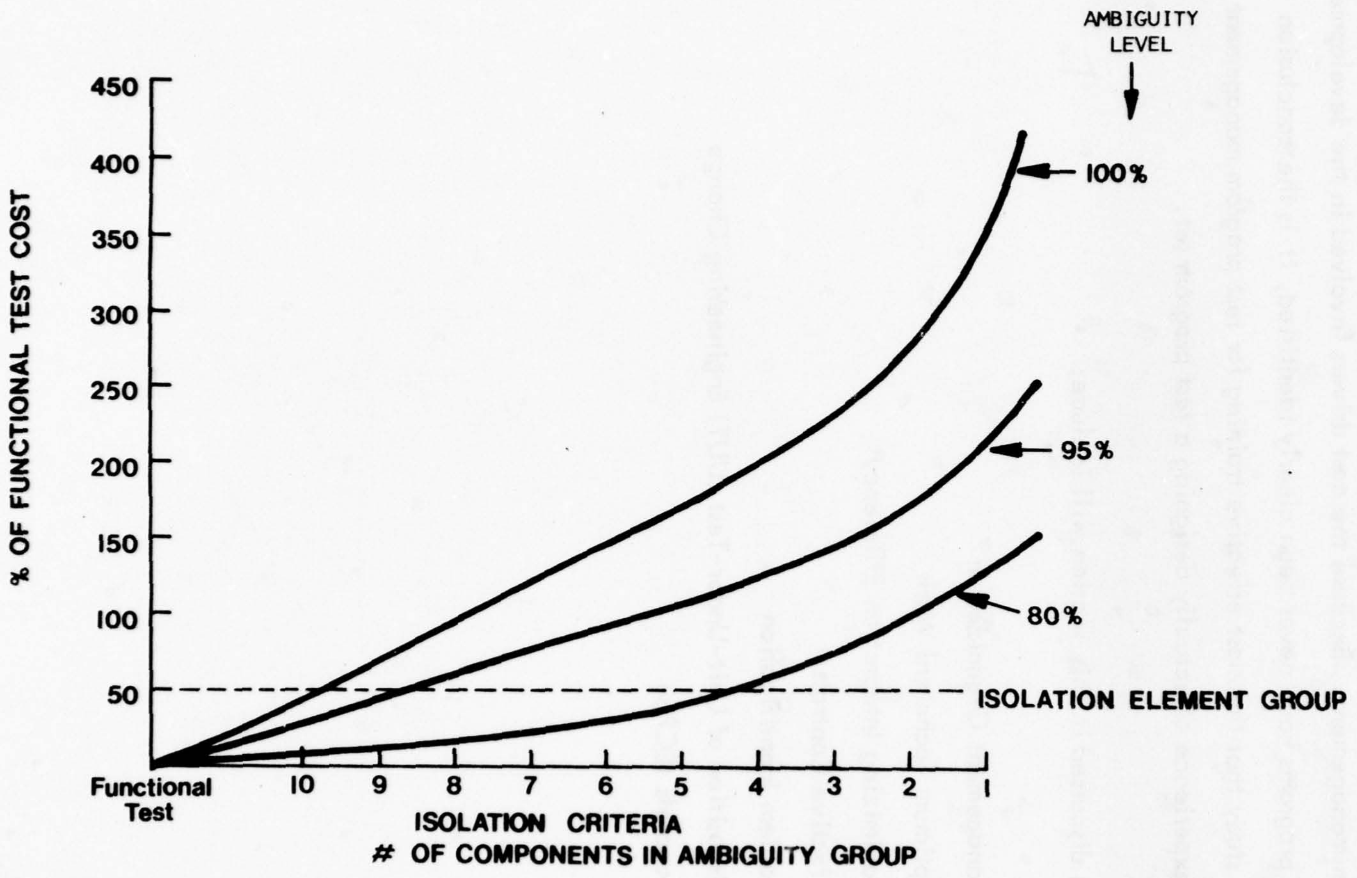


FIGURE 4

PERCENT OF FUNCTIONAL TEST COST VERSUS NUMBER OF COMPONENTS IN AMBIGUITY GROUP AND AMBIGUITY LEVEL.

MANAGEMENT CONTROLLED COST FACTORS

One of the most intangible, yet significant, factors relating to cost in a test program development program is the viability and experience of the program management. Because the cost drivers involved in the development of test programs have never been clearly identified, it is the conclusion of this study that the most effective training for test program management is the experience of actually designing a test program set.

Factors discussed in this section will include:

- Management Organization
- Optimum Technical Work
- Maximizing Integration Efficiency
- Effective Controls
- Problem Identification
- Disposition of Unit-Under-Test (UUT) Engineering Change Proposals (ECP's)

Management Organization

The development of a test program set for any electronic package is an engineering design task. Certainly, the first level of management must be technically oriented in order to handle the wide spectrum of technical problems that will be encountered. Because of the wide variety of support required in the development activity, the organization must be structured so that they can truly support rather than hinder programs. The support activities must be managed as if they were in competition for the work and services they provide. The support organizations should be separate, and evaluated on the quality of the service they provide.

Although test program set development includes many individual tasks such as circuit analysis, code and compile, program integration and customer sell-off, the organizations should not be set up to handle these tasks separately by different engineers for the same test program. It can be substantiated with data that was analyzed in this study that the most cost effective method of developing a test program is to have the same individual(s) perform all of the tasks necessary to produce the finished product. These cost savings can be from 25 to 75 percent of the analysis and test design hours. A good method of organizing the technical work would be along the complexity/electronic family type groups presented in the basic algorithm - RF, real-time digital, manual interface, etc. This allows the pooling of expertise within a particular electronic discipline and can create an air of performance competition among the different groups.

All management should dedicate themselves to the identification and removal of non-technical obstacles that will limit progress.

Optimum Technical Work Load

Because of the many support service factors that can result in lost time for the test engineer in the development of a test program set, the optimum technical work load for the test engineer is one LRU and one SRU, or three SRU's simultaneously. Ideally, if SRU programs are developed at the same time of the LRU programs, additional cost savings can be accomplished. The most cost effective way of developing LRU test programs is to develop programs for the SRU's first - it is unfortunate but our experience has shown that this has never been initiated on any major military program.

Maximizing Integration Efficiency

In most test program development activities, the demands on test equipment time create the most critical scheduling problems. It is recommended that four hours per day per man on the test equipment is the optimum from an efficiency standpoint. In determining station scheduling, the best workable solution is some form of a queuing concept. The staging and coordination of physical assets can be counterproductive when using a fixed schedule, wherein the queuing concept minimizes lost time. This concept can be implemented in many ways. A suggested method is:

> All requests for test station time are channeled through a station coordinator. In order to request time, the test engineer must possess an object program and program listing and identify the other assets - unit under test, interface adapter, etc., required to perform his integration work. On a first-come, first-served basis, the station coordinator allocates stations as they become available. Other support personnel collect all required assets for the test engineer so they are ready when the station becomes available. A technical floor manager monitors the progress of station activity and informs the station coordinator when stations are becoming available.

This method of station scheduling is effective and minimizes idle-stations.

Effective Controls

Technical management controls are important in the development of any computer software. The most important factors are discussed here.

Design reviews, if not properly structured, can result in wasted time for both the test engineer and the reviewers. The foremost fact is to establish if the engineer has a good functional understanding of the electronics to be tested and, if so, a logical review of his test strategy should be accomplished. Areas of uncertainty should be identified for continued monitoring and alternate test methods discussed. This type of review will generate meaningful feedback, prepare the test engineer for the integration phase and provide the supervisor with a better insight of what to look for in subsequent status reviews.

At all times, the configuration of the test equipment and all support software must be monitored, and controlled changes communicated to the test engineer.

The configuration control procedures for the test program set must be clear and enforced in order to maximize efficiency and minimize confusion. It is more efficient to allow the test engineer to maintain the configuration until the completion of the integrated functional test. From that configuration baseline, changes can create problems, and should be reviewed and approved by the technical supervisor for formal change approval.

Problem Identification

The key to the successful management of a test program development activity is the timely identification and solution of problems. A problem reporting system that is easy to understand, quick reacting and useful will achieve the best results. Integration anomalies are often postponed or covered up if not reported as they occur and solved in a systematic fashion.

Typical problems requiring investigation, communication or dissemination:

- o Non-repeatable functional test
- o Test system anomalies - hardware, software or documentation
- o Test system updates
- o Unit under test configuration errors
- o Tested module interchangeability problems

Disposition of Unit-Under-Test ECP's

Changes to the UUT and their effect on a test program are often confused and mishandled. ECP's normally fall into three categories:

1. Changes required to make the electronics perform to specification.

2. Changes made to improve reliability.

3. Changes adding new or expanded capability to the electronics.

Test program design is normally affected in some manner by all changes, even some that are originally thought to be Class II or compatability changes.

The most effective and efficient way to deal with changes of any kind is to develop the test program to a baseline and make the necessary changes later.

AUTOMATIC TEST PROGRAM GENERATION (ATPG)

Automatic Test Program Generation is being accepted as the way to produce test programs for digital modules. Current thinking and ATPG design efforts are expanding this philosophy to system testing as well.

The concept of ATPG is to model the circuit elements to create a simulation of the electronics on a digital computer. Test patterns are then inputted and executed on the simulation model to produce the corresponding response pattern. Fault isolation is accomplished in two different ways depending upon the ATPG implementation approach.

1. Failing outputs are traced backwards on the test equipment using manual probing. The simulator directs the test technician where to probe until the actual failure can be isolated to a component.

2. The other method is to force-fail all components, one at a time, in the simulation model recording the response patterns produced with each failed condition creating a fault dictionary.

Method 1 results in using more test station time for isolation of a failure, and method 2 requires large amounts of simulation time. Both methods have advantages in different situations.

Provided the modules are suited for ATPG, test program development costs will be reduced. ATPG also results in better test visibility because of the consistency of the simulation; however, each individual module should be reviewed before this decision is made.

1. Many modules that are single function in nature and simple in design should still be done manually.

2. One must also use caution in determining whether all elements of the module can be accurately modeled.

3. Most simulators cannot handle initialization efficiently.

4. Counters and other sequential logic situations require large amounts of simulation time.

5. Since ATPG systems to date are static logic designed, real-time failures might often go undetected. Modules that pass by themselves but fail in their system environment cause significant support problems. A careful review of the module's function in the system, and the system failure it can cause, should be made before static testing is selected as the method of support.

In many instances, a combination of manually generated tests and ATPG is the ultimate answer.

Data evaluated in this study showed that two (2) man-hours per component is the average cost of generating a complete test program on an interactive ATPG system with guided probe diagnostics. Of this cost, approximately 25% will be spent in creating the model and 75% on the test equipment verifying the test program. These numbers include a 20% rework factor to offset the human error element. If the ATPG system to be used is not interactive, the estimate should be six (6) man-hours per component (75% modelling).

Two factors not included in the cost estimate which have to be identified and estimated separately are as follows:

- Modules which have not been designed for easy initialization.
- Component models which are not contained in the ATPG software library.

The creation of a component model is the equivalent of generating a test program for the component and should be estimated accordingly.

Once the module has been verified on the test equipment, only minimal fault insertion is cost effective. Massive fault insertion only verifies the accuracy of the ATPG system itself which is more effectively accomplished by simulating a larger population of modules rather than concentrated emphasis on an individual module.

DEVELOPMENT VERSUS LIFE CYCLE COSTS

Often in the procurement of support for electronics, decisions are made which force the development costs of the life cycle cost of the equipments higher than required. These situations are normally driven by actual requirements or by "parochial opinions" and can often be avoided if certain trade-offs and compromises are evaluated. Some of the most cost critical decisions are discussed below:

On-Line Editing and Compilation

Within the past few years, technology in software development has advanced to the point where high level test languages such as ATLAS are available on today's test systems. With this source language capability on the test system, on-line editing and compilation at the test station are now possible during program integration. In the past, this was normally an off-line process on a larger non-dedicated computer. Study results reveal that the ability to perform these functions can effect a 40 percent cost reduction in the integration phase of a test program development. The ability to change a test program on the test station is undesirable for obvious reasons in a field maintenance activity and, therefore, sometimes omitted from the procurement requirements of the test system. It is a simple and inexpensive problem to delete this capability for the field activity.

Common Interface Hardware

Advantages to obtaining the maximum degree of common interface equipment for the field maintenance activities are well known - less outfitting costs, ease of transportability and storage, and fewer logistics requirements of the support equipment. When consideration is given to field test and repair activities, considerable cost savings can be realized.

An awareness must be given to the problems this places on the test engineer and scheduler during the development phase of the support. If several electronic equipments are using the same interface hardware, then it must be assumed that several test engineers will have to share this hardware for different development tasks. Other than the obvious problem of scheduling the integration activity, the largest cost escalator results from the change restrictions imposed on the engineer by this hardware sharing. In order to accomplish the integration activity effectively, it can be assumed that the engineers will be on different shifts, thereby complicating coordination and communication resulting in increased development costs.

Since test program development involves many changes, any unnecessary change restriction on the test engineer will escalate the development costs.

It is recommended that in the development of test programs, that each electronic device have its own development interface hardware. At the conclusion of the development cycle, an optimization design study should be performed and common hardware elements created at that time.

FAULT INSERTION AND CUSTOMER "SELL-OFF"

A major development program evaluated in this study incorporated an aggressive fault insertion plan for both program integration and customer sell-off. It is not intended that any of the conclusions based on the following discussion be incorporated into every program, but to point out the costs involved and alternate approaches that having been implemented might have been more cost effective.

One of the dilemmas in any software development program is the method and budget to be used in "proofing"/"validating"/"debugging" of the object program. Since the effectiveness of diagnostic software is measured by the accuracy of field detection and isolation of operational failures, much of the developed and delivered software is never used.

Data published by NAEC on a regular basis indicates operational failures of the VAST system over a large operating life. The latest report used in this study indicated that in over 600,000 operational hours, 30% of the modules in the system had never failed, and an additional 39% of the modules had failed less than five (5) times. Attempts were made to gather similar data for other military electronics systems, but a source could not be found. The point to be made is that in all of today's electronics, certain elements in the system will never fail. This is not to conclude that these modules will not be removed as part of a maintenance action, but that any developed diagnostic software would have never been utilized. Prediction of this type of information is never 100% accurate, but is at least a starting point for potentially large cost savings. In the early stages of any electronic system development, the weak areas are identified empirically and identified for reliability improvements. This then causes a change in the data base of failed elements and a different population appears.

With this background in mind, we will discuss the costs incurred in a large avionics development program. The monies spent in integration fault insertion and sell-off was approximately 250,000 man-hours. This cost included the insertion of 9,000 faults during program integration and 4,000 faults during customer sell-off of some 60 electronic assemblies.

Based upon this data, it is dubious that the extensive fault insertion and sell-off was the best financial decision in achieving effective field support for the electronics. In addition to the man-hours spent on diagnostic proofing, many thousands of dollars were spent in the repair of the accidental failures caused in the electronic units under test during this part of the program. In only a few instances did the fault insertion level ever reach 100% of the possible failures, so even after the expenditures made, diagnostic errors were discovered in the field.

Because of factors mentioned earlier, it has been concluded that good functional test programs with an ongoing software diagnostic maintenance activity would have been more cost effective. Without extensive fault insertion, most field failures could have been detected and correctly isolated. Those that were incorrect could have been resolved by the diagnostic software maintenance team. This would have allowed more budget for electronic assembly spares and concentrated the diagnostic software dollars on actual field failures.

It is suggested that a small team of test engineers select a small number of critical faults for each UUT. This team should concentrate on faults that have a high probability of occurrence, not a random sample of the diagnostic accuracy.

Another suggestion is to contract for diagnostic software warranty at the outset of test program development. Although this would probably be very costly for the first few test cases, overall life cycle costs could be greatly reduced.

COMMERCIAL VERSUS MILITARY SUPPORT

Although many differences exist in the two different support worlds, some of the more thought-provoking are discussed here for future consideration.

Military philosophy has been to repair all types of failures at either the field support or depot facility. At military training facilities, some technicians are taught the repair of multi-layer modules even where failures have caused damage to several layers of etch. Some maintenance training includes the repair of hybrid circuit components themselves. Although this is very admirable, it is rarely done in the commercial world because of the high costs involved. Labor dollars are no longer free in the military, and the technology and tools required are very expensive.

Many commercial diagnostic philosophies use the guided probe approach to digital test isolation. This reduces test program development costs and produces a higher rate of repair on failed boards. The problems of guided probing on the conformally coated military modules should be concentrated on so that these cost advantages can be realized.

Commercial companies can only survive if they can effectively support their products in the field and maintain good consumer relations. Failures at most commercial installations, wherever they may be located, are normally corrected in a matter of hours and in the worse case days. This is accomplished by using well trained field representatives with appropriate and timely access to module spares.

TEST CASES

The LRU selected as a test case was the Radio Receiver R-1849( )/ULR-17(V). This HF receiver is capable of receiving and demodulating AM, FM, CW/FSK and SSB (USB, LSB) signals over the frequency range of 0.5 to 30 MHz. The receiver can provide up to eight phase and gain matched channels.

The radio has 69 input/output pins which were placed in the following categories.

12	Manual Interface
17	RF
40	Other

The RF circuitry was not accessible, so the accessibility Factor of 1 was used on the RF pins.

The LRU was run on the algorithm's FORTRAN program (Appendix B) with the following options, and results.

Unmature EQUATE, (t) = 1
2844 man-hours
425 station-hours

Mature EQUATE, inexperienced man
2419 man-hours
340 station-hours

Mature EQUATE, experienced man
2256 man-hours
307 station-hours

Mature EQUATE without on-line edit, experienced man

2983 man-hours

452 station-hours

Unmature EQUATE without on-line edit

3572 man-hours

570 station-hours

The SRU selected was the Analog Scanner C Circuit Card. The schematics for the circuit card are attached as Appendix C.

The test program for the SRU was estimated using three (3) different approaches.

Using interactive ATPG

47 modeled components x 2 = 94 man-hours

70 station-hours required

Non-interactive ATPG

47 modeled components x 6 = 282 man-hours

70 station-hours required

Manual generation from algorithm

75 static pins @ 5 hours/pin = 375 man-hours

37.5 station-hours required

SUMMARY/CONCLUSION

The results of this study were totally derived from empirical data generated from previous programs. As such, the study should be updated periodically with new data base information as it is generated to update the algorithm.

As is self evident, the advancement of a highly specialized community of Automatic Test Engineers will tend to drive prohibitive costs downward due to management techniques, learning curve growth, and by the incorporation of many of the suggestions outlined in this study.

The algorithm was developed from the sampled electronics creating a "norm" average for each category. As depicted in Table II, some of the sampling densities were not of a large population. Therefore these items, i.e., microprocessor, should be evaluated judiciously until ECOM's data base is larger.

In conclusion, DSII believes that the application and usage of the algorithm will provide ECOM with a valuable estimating tool in assessing costs associated with Test Program Set (TPS) developments.

APPENDIX A

List of Equipments Evaluated as Part of the Data Base

Radar Receiver Transmitter	RT1023/APN-201
Height Indicators	ID-1770/APN-201
Air Navigation Multiple Indicators	ID-1779/A
Attitude-Course-Height Deviation Indicators . . .	ID-1780/A
Navigation Data Converter-Repeater	CV-2854/A
Airspeed-Altitude Computer	CP-1077/AYN-5
Radar Navigation Set	AN/APN-200
Sonobuoy Bearing-Range Receiver	R-1768/ARS-2
Navigation Data Converter	CV-2745/ASA-84
Navigation Control	C-8746/ASA-84
Displacement Gyroscope	CN-1366/ASN-107
Analog to Digital Converter	CV-2858/ASN-107
Rate Gyroscope Assembly	CN-1370/ASW-33
Flight Data Computer	CP-1074/ASW-33
DTS Digital to Analog Converter	CV-2830/AYC
Antenna Coupler	CU-1985/ARC-153
Radio Frequency Amplifier	AM-6384/ARC-153
Radio Receiver-Transmitter	RT-1016/ARC-153
Radio Receiver-Transmitters	RT-1017/ARC-156
Converter-Interconnecting Box	CV-3048/AI
Intercomm-Comm Control Group Control	C-8760/AI
Intercommunication Stations	LS-601/AI
Command Signals Decoder-Programmer	KY-747/ASQ-147
Power Supply	PP-6664/ASQ-147
Armament Subsystem Control	C-8857/ASQ-147
Command Signals Decoders	KY-745/ASQ-147
Command Signals Decoder	KY-746/ASQ-147
Bomb Bay Distribution Box	J-3069/ASQ-147

APPENDIX A (Cont'd)

Signal Data Converter	CV-2882A/AYS
Signal Generator-Spectrum Analyzer	SG-962A/AYS
Spectrum Analyzer Converter	CV-2883A/AYS
Sonar Data Computer	CP-(1094)AYS
Sonobuoy Monitor Panels	58 SB-3593/AYS
Data Storage Magnetic Drum	MU-576/AYS
Power Supply	PP-6671/AYS
Radio Frequency Amplifiers	AM-6418/ARR-76
Radio Receiver	R-1741/ARR-76
Magnetic Tape Transport	RD-349/ASH-27
Tape Transport Interface Unit	MX-8959/ASH-27
Power Supply (No. 1)	PP-6679(P)AYK-10V
Power Supply [CP]	PP-6675/AYK-10(V)
Power Supply [IO]	PP-6677/AYK-10(V)
Power Supply [MEM]	PP-6676/AYK-10(V)
Tactical Indicator Control (Pilot)	C-8862/ASQ-147
Computer-Indicator Control (Co-Pilot)	C-8859/ASQ-147
Computer Indicator (TACCO)	C-8860/ASQ-147
Digital to Analog Converter	CU-2806/ASA-82
Tactical Indicator [Pilot Display]	IP-1051/ASA-82
Tactical Indicator [Co-Pilot MPD]	IP-1053/ASA-82
Tactical-Acoustic Indicators	IP-1054/ASA-82
Acoustic Indicator [ARU]	IP-1052/ASA-82
Signal Data Recorder-Reproducer	RD-348/ASH
Analog to Digital Converter	CV-2881/AS
Power Supply-Video Converter	PP-6611/AA
Infrared Control-Converter	C-8759/AA

APPENDIX A (Cont'd)

Countermeasures Receiver	R-1742/ALR-47
Signal Comparator	CM-416/ALR-47
Radar Set Control	C-8788/AP

```
FORTRAN IV      V01C-03F    FRI 31-MAR-78 15:48:43       PAGE 001

      DIMENSION  ICAT(6),PINS(6),IACES(6),CATG(8),ACCES(8),
     1ATEM(18),ALC(18),EXP(10)
      DATA CATG /40.,35.,30.,20.,10.,5.,20.,10./
      DATA ACCES /1.00,1.00,.670,0.0,0.0,1.00,.500,0.0/
C     INPUT THREE CARDS
      READ(4,1000) ITPS,IED,TK ,TIME
 1000 FORMAT(I1,X,I1,X,F2.0,X,F2.0)
      READ (4,1010) (ICAT(J),PINS(J),IACES(J),J=1,6)
 1010 FORMAT(6(I1,X,F3.0,X,I1,X))
      READ(4,1020) (EXP(J),J=1,10)
 1020 FORMAT(26(F2.0,X))
      ICNT=1
      HRS=0
      K=1
      SMAN=0
C     GET NORMALIZED MAN HOURS
 1030 ICAT1=ICAT(ICNT)
      SMAN=PINS(ICNT)*CATG(ICAT1)
      HRS=HRS+SMAN
      ICNT=ICNT+1
      IF(PINS(ICNT).NE.999)GO TO 1030
C
C     TRA ONLY  PRINT SULTS
      TRA= .45*HRS
      GRAT= .55*HRS
      IF(ITPS.NE.2)GO TO 1050
      WRITE(6,1040) TRA
      GO TO 1400
 1040 FORMAT(16H1TRA MAN HOURS= ,F6.0)
C     IF ACCES IBLILITY IS A PROBLEM TAKE % FROM TABLE ADD TO INTG.
 1050 IF (IACES(K).NE.0)GO TO 1055
      ICAT1=ICAT(K)
      X=GRAT*ACCES(ICAT1)
      GRAT=GRAT+X
 1055 K=K+1
      IF(K.LT.ICNT)GO TO 1050
C
C     ON LINE EDIT AVAILABLE,IF NOT ADD 40% TO INTEGRATION
C
C
 1060 IF(IED.EQ.1)GO TO 1070
      X=.4*GRAT
      GRAT=GRAT+X
 1070 GO TO (1100,1100,1080,1090)ITPS
C
C     DID ONE ENGR DO ALL TASKS? IF YES GO TO MANPOWER CALCULATION
C     TRA DONE BY SAME VENDOR
 1080 X=.75*TRA
      TRA=TRA-X
      GO TO 1100
C     TRA DONE BY DIFFERENT VENDOR
 1090 X=.25*TRA
      TRA=TRA-X
C     GET TOTAL MAN-HOURS AT THIS POINT
 1100 HRS=TRA+GRAT
```

```
FORTRAN IV      V01C-03F   FRI 31-MAR-78 15:48:43                    PAGE 002

C      MORE THAN ONE ENGR REQUIRED?
       IF(HRS.GT.2988.)GO TO 1110
C      NO
       SMOS=HRS/166
       SMEN=1.
       WRITE(6,1105)HRS,SMOS
 1105  FORMAT(8H LINE11=,F5.0,2HM=,F3.0)
       GO TO 1120
 1110  SMEN=HRS/2988
C      NUMBER OF MEN FOR 18 MOS.
       EXMOS=(HRS-(SMEN*2988.))
       IF(EXMOS.GT.0)SMEN=SMEN+1.
       SMOS= HRS/SMEN
 1120  ICNT=1
C
C1130  TEM=(100.*TIME**2.)+(7.08*TK-220.8)*TIME+(0.15*TK**2.+24.5*TK)/
C     1TIME**2.+(0.0708*TK-2.208)*TIME+(0.15*TK+24.5)
 1130  TS=100.*TIME**2.
       TJ=(7.08*TK-220.8)*TIME
       TI=.15*TK**2.+24.5*TK
       TT=TS+TJ+TI
       SI=TIME**2.
       SS=(0.0708*TK-2.208)*TIME
       ST=0.15*TK+24.5
       STOT=SS+ST+SI
       TEM=TT/STOT
C        CALCULATE MATURITY
       IF(TEM.GE.90.)GO TO 1210
       ATEM(ICNT)=TEM
       ICNT=ICNT+1
       TIME=TIME+1
C
       GO TO 1130
 1210  TEM=90.
 1220  ATEM(ICNT)=TEM
       ICNT=ICNT+1
       IF(ICNT.LE.SMOS)GO TO 1220
C      OUTPUT MATURITY
       ICNT=1
 1230  WRITE(6,1240)ICNT,ATEM(ICNT)
 1240  FORMAT(27H TEM FOR DEVELOPMENT MONTH ,I2,2H+=,F5.2)
       ICNT=ICNT+1
       IF(ICNT.LE.SMOS)GO TO 1230
       TMEN=1
 1250  ICNT=1
 1255  EXM=EXP(TMEN)
 1256  X=1+(ATEM(ICNT)/90)
C      TEM=100.*EXM**2.+(19.*TK*X+585.)*EXM+(40.5*TK*X+2835.)/
C     1EXM**2.+(0.161*TK*X+5.65)*EXM+81.
       TS=100.*EXM**2.
       TJ=(19.*TK*X+585.)*EXM
       TI=40.5*TK*X+2835.
       TOT=TS+TJ+TI
       ST=EXM**2.+(0.161*TK*X+5.65)*EXM+81.
```

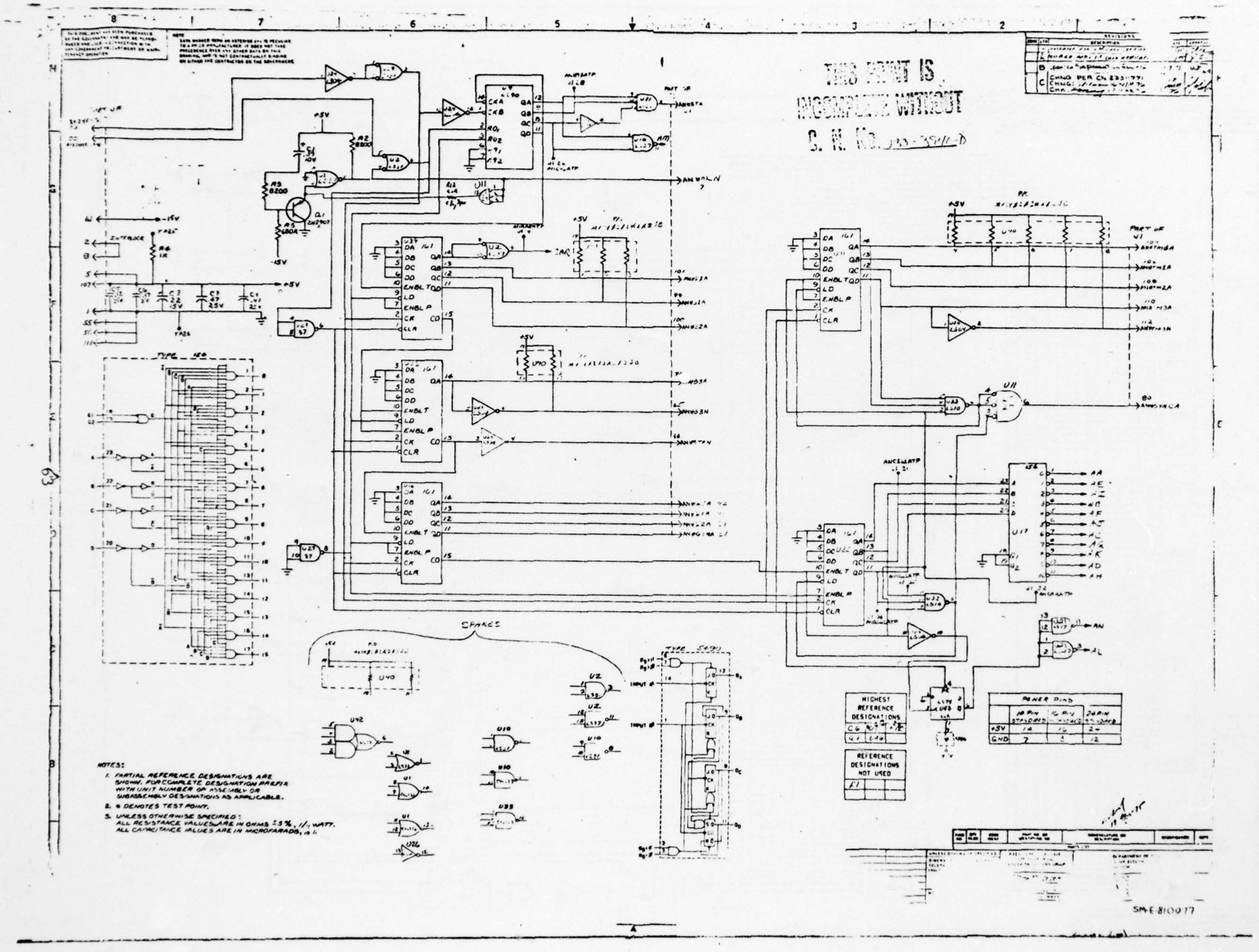
```
FORTRAN IV      V01C-03F   FRI 31-MAR-78 15:48:43                 PAGE 003

             TEM=TOT/ST
                WRITE(6,1260)TMEN,ICNT,TEM
             IF(TEM.GE.100.)GO TO 1265
             ALC(ICNT)=TEM
       1260  FORMAT(25H LEARNING CURVE FOR MAN #,F2.0,7H MONTH ,I2,2H =,F4.0)
             THRS=(100.-(ALC(ICNT)*ATEM(ICNT))/100)*1.66
             GRAT=GRAT+THRS
             HRS=HRS+THRS
             EXM=EXM+1
             ICNT=ICNT+1
             IF(ICNT.LT.SMOS)GO TO 1256
       1265  TEM=100.
             ALC(ICNT)=TEM
             WRITE(6,1260)TMEN,ICNT,ALC(ICNT)
             THRS=(100.-(ALC(ICNT)*ATEM(ICNT))/100.)*1.66
             IF(THRS.LT.0.)GO TO 1266
             GRAT=GRAT+THRS
             HRS=HRS+THRS
             IF(ICNT.GE.SMOS)GO TO 1266
             ICNT=ICNT+1
             GO TO 1265
       1266  TMEN=TMEN+1
             IF(TMEN.LE.SMEN)GO TO 1250
             SMOS=HRS/166
             IF(SMEN.LE.1.)GO TO 1285
              SMOS=SMOS/SMEN
       1285  STHRS=.2*GRAT
             WRITE(6,1290)HRS,SMOS,SMEN,STHRS
       1290  FORMAT(22H1TOTAL HOURS REQUIRED ,F5.0,15H  DEV MONTHS = ,F3.0,
            1 18H  NUMBER OF MEN = ,F3.0,13H STATION HRS=,F6.2)
       1400  STOP
             END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET  ATTRIBUTES

ICAT     000006  INTEGER*2 ARRAY (6)
PINS     000022  REAL*4    ARRAY (6)
IACES    000052  INTEGER*2 ARRAY (6)
CATG     000066  REAL*4    ARRAY (8)
ACCES    000126  REAL*4    ARRAY (8)
ATEM     000166  REAL*4    ARRAY (18)
ALC      000276  REAL*4    ARRAY (18)
EXP      000406  REAL*4    ARRAY (10)
ITPS     001160  INTEGER*2 VARIABLE
IED      001162  INTEGER*2 VARIABLE
TK       001164  REAL*4    VARIABLE
TIME     001170  REAL*4    VARIABLE
J        001174  INTEGER*2 VARIABLE
ICNT     001176  INTEGER*2 VARIABLE
HRS      001200  REAL*4    VARIABLE
K        001204  INTEGER*2 VARIABLE
SMAN     001206  REAL*4    VARIABLE
ICAT1    001212  INTEGER*2 VARIABLE
TRA      001214  REAL*4    VARIABLE
GRAT     001220  REAL*4    VARIABLE
X        001224  REAL*4    VARIABLE
SMOS     001230  REAL*4    VARIABLE
SMEN     001234  REAL*4    VARIABLE
EXMOS    001240  REAL*4    VARIABLE
TS       001244  REAL*4    VARIABLE
TJ       001250  REAL*4    VARIABLE
TI       001254  REAL*4    VARIABLE
TT       001260  REAL*4    VARIABLE
SI       001264  REAL*4    VARIABLE
SS       001270  REAL*4    VARIABLE
ST       001274  REAL*4    VARIABLE
STOT     001300  REAL*4    VARIABLE
TEM      001304  REAL*4    VARIABLE
TMEN     001310  REAL*4    VARIABLE
EXM      001314  REAL*4    VARIABLE
TOT      001320  REAL*4    VARIABLE
THRS     001324  REAL*4    VARIABLE
STHRS    001330  REAL*4    VARIABLE
```



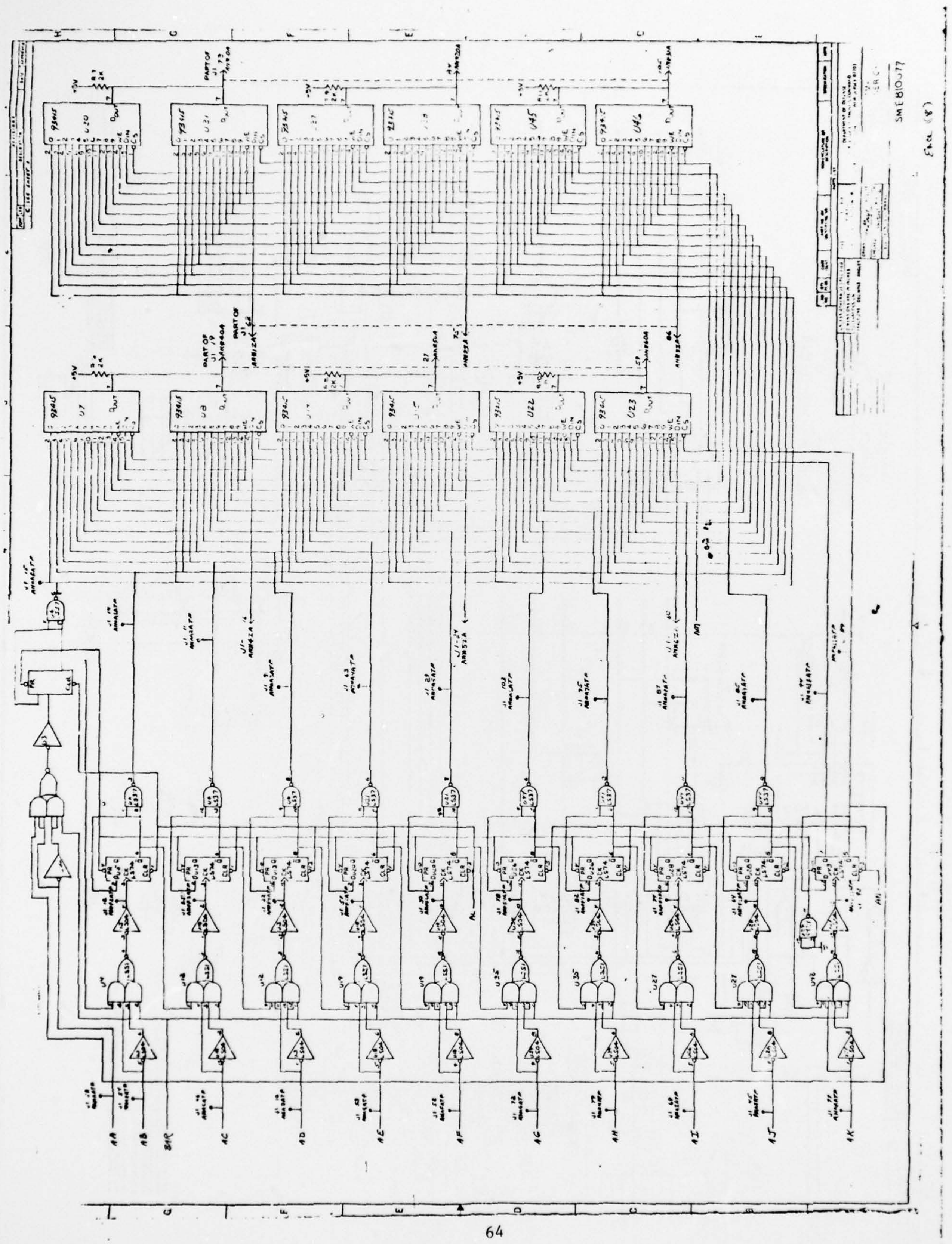
HIGHEST REFERENCE DESIGNATIONS		
C6	[illegible]	[illegible]
Q1	U40	

REFERENCE DESIGNATIONS NOT USED		
R1		

POWER PINS			
	14 PIN STANDARD	16 PIN STANDARD	24 PIN STANDARD
+5V	14	16	24
GND	7	8	12

NOTES:

1. PARTIAL REFERENCE DESIGNATIONS ARE SHOWN. FOR COMPLETE DESIGNATION PREFIX WITH UNIT NUMBER OR ASSEMBLY OR SUBASSEMBLY DESIGNATIONS AS APPLICABLE.
2. ● DENOTES TEST POINT.
3. UNLESS OTHERWISE SPECIFIED:
   ALL RESISTANCE VALUES ARE IN OHMS ±5%, 1/4 WATT.
   ALL CAPACITANCE VALUES ARE IN MICROFARADS, 10%.



HISA
FM 1183-79